MASSIVELY PARALLEL NEUROCOMPUTING FOR AEROSPACE APPLICATIONS

Amir Fijany Jacob Barhen Nikzad Toomarian

Jet I'repulsion Laboratory
California Institute of Technology
Pasadena, CA 91109, USA

ADSTRACT

An innovative hybrid, analog-digital charge-domain technology, for the massively parallel VLSI implementation of certain large scale matrix-vector operations, has recently been introduced. It employs arrays of Charge Coupled/Charge Injection Device cells holding an analog matrix of charge, which process digital vectors in parallel by means of binary, non-destructive charge transfer operations. The impact of this technology on massively parallel processing is discussed. Fundamentally new classes of algorithms, specifically designed for this emerging technology, as applied to signal processing, are derived.

1. INTRODUCTION

The ever expanding capabilities of space missions' data acquisition systems is creating an unprecedented gap in our ability to process the underlying massive volume of information. Thus, our goal at this paper is to present a fundamentally new classes of algorithms, specifically designed for emerging compact integrated systems composed of salable, high performance, massively parallel computing units in charge-domain technology.

Many algorithms required for scientific modeling will make frequent use of a few well defined, often functionally simple, but computationally very intensive data processing operations, Those operations generally impose a heavy burden on the computational power of a conventional general-purpose computer, and run much more efficiently on special-purpose processors that are specifically tuned to address a single intensive computation task only.

An innovative hybrid, analog-digital charge-domain technology, for the massively parallel VLSI implementation of certain large scale matrix-vector operations, has recently been developed and applied at Caltech [1,2.3,5,6], It employs arrays of Charge Coupled/Charge Injection Device (CCD/CID) cells holding an analog matrix of charge, which process digital vectors in parallel by means of binary, non-destructive charge transfer operations.

Figure 1 shows a simplified schematic of the CCD/CID array. Each cell in the array connects to an input column line and an output row line by means of a column gate and a row gate. Both gates together hold a charge in the silicon substrate underneath them that represents an analog matrix element. By default, the matrix charge sits under the column gates. In the basic matrix-vector multiplication (MVM) mode of operation, for binary input vectors, the column gates serve as binary-analog multipliers by transferring the matrix charge towards the row gates *only* if the input bit of the column indicates a binary 'one'. The charge transferred under the row gates is summed capacitively on each output row line, yielding an analog output vector which is the product of the binary input vector with the analog charge matrix. By virtue of the CCD/CID device physics, the charge sensing at the output is of a non-destructive nature, and the charge matrix is restored in its original state simply by pushing the charge back under the column gates.

A bit-serial digital-analog MVM can be obtained from a sequence of binary-analog MVM operations, by feeding in successive vector input bits sequentially, and adding the corresponding output contributions after scaling them with the appropriate powers of two. A simple parallel array of divide-by-two circuits at the output accomplishes this task [3,6].

The particular choice of this unusual charge-domain technology resulted from several considerations not just limited to issues of speed and parallelism. In comparson to other, more common parallel high-speed technology environments (digital CMOS, etc.), the distinct virtues of our charge-domain technology for large-scale special-purpose MVM operations are the following:

Very High Density: The compactness of the CCD/CID cell allows the integration of up to 10^5 cells on a $1cm^2$ die (in a standard 2pm CMOS technology), providing single-chip 100 GigaOPS computation power.

Very Low Power Consumption: The charge stored in the matrix is conserved along the computation process because of the non-destructive nature of the CCD/CID operation. Hence, the entire power consumption is localized at the interface of the array, for clocking, 1/0 and matrix refresh purposes. This enables the processor to operate at power levels in the mW/TeraOPS range.

Scalability: The salable architecture of the CCD/CID array allows the interfacing of many individual processors in parallel, combining together to form effective processing units of higher dimensionality, still operating at nominal speed.

1/0 Flexibility: Although an analog representation is used inside the array to obtain fast parallel computation, the architecture of the processor provides the flexibility of a full digital interface, eliminating the bandwidth and noise problems typical for analog interfacing.

Programming Flexibility: The architecture allows for either optical (parallel, sustained [1]) or electronic (semi-parallel, periodical [6]) loading of the charge matrix. The latter method requires interrupts, of duration usually much shorter than the time interval in computation mode, for which the stored charge matrix remains valid before a matrix refresh is needed.

A variety of MVM processors of different dimensions have been designed and tested at Caltech. Preliminary results on a 128 x 128 working prototype, implemented on a single 4mm x 6mm die in $2\mu m$ CCD-CMOS technology, indicate a performance of approximately 10^{10} 8-- bit multiply-accumulates per second. Processors with 1024 x 1024 cells will be realizable in the mar future.

2, IMPACT ON PARALLEL COMPUTATIONAL COMPLEXITY

An innovative approach to parallel algorithms design remains a key enabling factor for high performance computing. In contrast to conventional approaches, one must develop computational schemes which exploit, from the onset, the concept of massive parallelism. The CCD/CID neural chip represents a promising hardware technology for massively parallel computation, which offers both opportunities and challenges in the design of parallel algorithms. This new technology *defies some of the* most baste *assumptions* in the analysis of computat ional complexity of parallel algorithms. '1'0 clarify this, a short discussion is needed. in what follows, N x 1 vectors and N x N mat rices are considered. Also, m and a stand for multiplication and addition, respectively,

One of the most fundamental results regarding the complexity of parallel computation addresses the summation of N scalars [7,8].

Theorem 1. The summation of N scalars can be performed in $O(\log N)a$ operations with $O(N)$ processors, Hence,

Corollary 1. A matrix-vector multiplication can be performed in $O(\log N)a + 0(1)m$ operations with $0(N^2)$ processors.

interestingly, the result of Theorem 1 has been assumed to be independent of hardware technology. This reflects a basic feature of digital computers, which do not allow the simultaneous summation of N numbers, The CCD/CID technology defies this most basic assumption, since the summation of N numbers is just the summation of N charges, which can be performed simultaneously. This is a *fundamental* change, which leads to a new set of results regarding the complexity of parallel computation using CCD/CID architectures. We submit that:

Theorem 2. The summation of N scalars can be performed in $O(1)a$ operations with $0(N)$ "processors" (CCD/CID cells). '1'bus,

Corollary 2. A matrix-vector multiplication can be performed in $O(1)a + 0(1)m$ operations with $0(N^2)$ '(processors" (CCD/CID cells).

It is important to note that the complexity of these operations is *independent* of the problem size, assuming that the chip size matches the problem.

The above results are not only significant from a theoretical point of view, since they improve the known time-lower bounds in these computations, but are also of practical importance. They show that the design and analysis of parallel algorithms based on CCD/CID neural architectures is drastically different from that for conventional parallel computers. The CCD/CID chip is currently considered for performing Matrix-Vector Multiplication (MVM) for which it achieves the computation time given in Corollary 2. However, in order to output the results, k clock cycles are needed, where k is the required number of bits. The CCD/CID chip

is most efficient for MVM computations wherein the matrix is known a priori; also of relevance are series of MVMs performed with the same matrix. Despite this seemingly narrow function, the CCD/CID neuroprocessor can be used **for** many applications, leading to new results.

3. SIGNAL PROCESSING APPLICATION

The foundation of conventional signal processing algorithms is based on the use of fast techniques for performing various discrete transformations such as I) FT, 1) S'1', DCT, DHT, etc... Consider the discrete Fourier Transform (DFT). The DFT can be represented by a Matrix-Vector Multiplication (MVM) with a computational complexity of $O(N^2)$. However, for both serial and parallel computation on conventional hardware, the Fast Fourier Transform (FFT) is always preferred. For serial computation, the FFT achieves a computational complexity of O(N $Log N$). Also, for implementation on parallel and vector computer architectures, the FFT has been considered as the base line algorithm. In particular, with O(N) processors, a time lower bound of 0(Log N) can be achieved in computing the FFT. Note, however, that this resu It is more of a theoretical importance than a practical one since, particularly for large N, implementat ion of the algorithm to achieve the above time lower bound would require an architecture with an excessive number of processors, and, more importantly, a very complex processors interconnection structure.

As stated before, with conventional hardware technology, the time lower bound in computing a MVM is 0(Log N) by using $O(N^2)$ processors. Note, however, that this result is more relevant to theory than to practice, since such an implementation of MVM requires a very complex parallel architecture. In contrast, a *practical* implementation of MVM on the CCD/CID chip can be performed in O(1) steps. Such a result indicates that, for **efficient** implementation of signal processing applications on CCD/CID chips, a new algorithmic framework is required, which significantly differs from the conventional ***fast*** *techniques* framework. In particular, the DHT, which **was** one of the topics investigated in **this task,** can be more efficiently implemented than the FHT. In fact, while the DHT can be performed in 0(1) with one CCD/CID chip, the implementation of FHT requires $O(Log$ N) chips and takes $O(Log$ N) steps!

In the following, we first discuss how a complex DFT can be obtained from the DHT by using CCD/CID chips. Then, a more interesting result is presented, regarding the convolution of two signals. We show that, by using CCD/CID chips, the convolution can be presented as a single MVM, and hence can be performed with the highest area and time efficiency, We also present a technique for area and time efficient DFT where the size of transform is much larger than the size of the CCD/CID chip. We conclude by outlining possible architectures that should enable high precision computing with charge-domain devices. Most of the ideas are further expanded in a recently published article, attached in appendix,

3,1 Computing DFT from DHT

We have considered the DHT as the main kernel for implementation on the CCD/CID chip. The DHT and its inverse are given [9, 1 O] as

$$H_{nm} = \frac{1}{N} \text{ cas } \frac{2\pi nm}{N} \tag{1}$$

and

$$H_{nm}^{-1} = N H_{nm} = \text{ cas } \frac{2\pi nm}{N} \tag{2}$$

the kernel for DHT and inverse DHT are shown in Figures 2 and 3 respectively,

In the above expressions

$$\text{Cas } (\theta) = \text{Cos } (\theta) + \text{ sin } (\theta) \tag{3}$$

The first issue is the computation of the DFT from the DHT. Let the DFT and DHT of a signal g be given as $f = F[g]$ and $h = H[g]$ where F and H are the DFT and DHT operators. In terms of the real and imaginary part of f it follows that

$$f = Re[f] + i\, Im[f] = E[h] - iO[h] \tag{4}$$

where E and O are even and odd operators defined as

$$E[h] \quad \frac{h^+ + h^-}{2} \rightarrow E[h_n] = \frac{h_n + h_{-n}}{2} = \frac{h_n + h_{N-n}}{2} \tag{5}$$

$$O[h] \quad \frac{h^+ - h^-}{2} \rightarrow O[h_n] = \frac{h_n - h_{-n}}{2} \cdot \frac{h_n - h_{N-n}}{2} \tag{6}$$

Let h^+ denote the vector h with normal ordering of the elements, i.e.,

$$h^+ = [h_0, h_1, \cdots h_{N-1}]^t \tag{7}$$

Also, let h denote the permutation vector h as

$$h^- = [h_0, h_{N-1}, \ldots h_1]^t \tag{8}$$

We note that h^+ and h^- can be related through a permutation matrix P as

$$h^- = Ph^+ \tag{9}$$

where P is given as

$$P = \begin{vmatrix} 1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 1 \\ \vdots & \vdots & \iddots & \vdots \\ 0 & 1 & \cdots & 0 \end{vmatrix}$$

From Eqs. **(5), (6)** and **(9),** the expressions for $E[h]$ and $O[h]$ are obtained as

$$E[h] = \frac{P+I}{2}h \quad \text{and} \quad O[h] = \frac{P-I}{2}h \tag{lo}$$

From Eqs. **(4)** and (10), f is obtained as

$$f = \frac{P+I}{2}h + i\frac{P-I}{2}h \tag{11}$$

$$f = \frac{P+I}{2}Hg + i\frac{P-I}{2}Hg \tag{12}$$

The matrices $\frac{P+I}{2}H$ and $\frac{P-I}{2}H$ are constant and can be precomputed. Therefore, Eq. (12) leads to a direct realization of DFT, as shown in Figure 4.

3.2 **Fast Area-Efficient Convolution Using DHT**

We consider the convolution problem, that is, the computation of a signal g as

$$g = {}^1g * {}^2g \tag{13}$$

where it is assumed that ${}^1g$ is the input and ${}^2g$ is known a priori, and hence its DHT can be precomputed. Let

$${}^1f = F[{}^1g] \text{ and } {}^1h = H[{}^1g] \tag{14}$$

$${}^2f = F[{}^2g] \text{ and } {}^2h = H[{}^2g] \tag{15}$$

$$f = F[g] \text{ and } h = H[g] \tag{16}$$

where F and H denote the DFT and DHT operators, respectively.

In accordance with Parseval's Theorem, we have

$$g = F^{-1}\{{}^1f \odot {}^2f\} = F^{-1}\{F[{}^1g] \odot F[{}^2g]\} \tag{17}$$

where F^{-1} is the inverse DFT operator and $\odot$ indicates component-by-component multiplication of two vectors. The above expression forms the basis of conventional convolution algorithms. Assuming 2_f to be known a priori, a first FFT is used to compute ${}^1f$ in $O(N LogN)$ steps, then the product of ${}^1f \odot 2_f$ is obtained in O(N) steps, and finally another FFT is used to compute f in $O(N LogN)$ step.

However, Eq. (17) is not suitable for implementation on CCD/CID chips. Hence, a new formulation is needed to take into account the fact that the only operation that can be efficiently performed is a MVM. In terms of DHT, h can be expressed[1 1] as

$$h = \frac{1}{2}[{}^1h^+ \odot {}^2h^+ - {}^1h^- \odot {}^2h^- + {}^1h^+ \odot {}^2h^- + {}^1h^- \odot {}^2h^+] \tag{18}$$

One can also express h in terms of even and odd operators as

$$\begin{aligned} h &= E[{}^2h^+] \odot {}^1h^+ + O[{}^2h^+] \odot {}^1h^- \\ &= E[{}^2h^+] \odot {}^1h^+ + O[{}^2h^+] \odot [P\ {}^1h^+] \end{aligned} \tag{19}$$

The $\odot$ operation of two vectors can be described by a diagonal matrix-vector multiplication as

$$v = {}^1v \odot {}^2v \rightarrow v = {}^1\hat{v}\ {}^2v \tag{20}$$

where ${}^1\hat{v}$ is a diagonal matrix whose diagonal elements are those of vector ${}^1v$. Using such a representation, Eq. (18) can now be expressed rra

$$h = \{\hat{E}[{}^2h^+] + \hat{O}[{}^2h^+]P\}\,{}^1h^+ \tag{21}$$

Let us define a matrix Q in terms of the matrices $\hat{E}$ and $\hat{O}$:

$$Q = \{\hat{E}[^2h^+] + \hat{O}[^2h^+]P\} \quad (22)$$

Note that, since it is assumed that ^2g is known a priori, the matrix Q is also known a priori. From Eqs. (]9) and (20), it follows that

$$h = QH^{-1}g \quad (23)$$

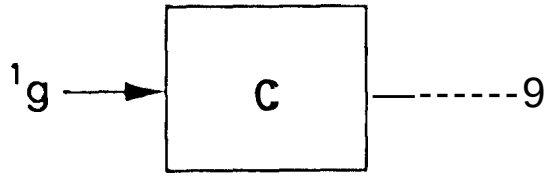
and finally

$$g = H^{-1}h = NHQH^{-1}g \quad (24)$$

Eq. (22) illustrates that convolution can be performed in terms of a simple matrix-vector multiplication, by defining an appropriate convolution operator. Specifitally, wc set

$$C = NHQH \quad (25)$$

This is illustrated below.



3.3 Computing Large DFT with Small Size CCD/CID Chips

We now consider the case wherein the size of the desired transform, N, is larger than the size of the CCD/CID chip, M. A direct ("brute force") approach would be to build an N XN DHT **matrix by using** M x M CCD/CID chips. **From our discussion in Sec.** 3., it then follows that 2($N/M)^2$ **CCD/CID chips of** size M x M would be required **for computation of a** DFT of size N.

The main **issues in devising a better approach** for performing large DFT with **small size** CCD/CID chips can be summarized **as** follows:

1. Preserve the computational efficiency;

2. Reduce the number of chips;

3. Reduce the complexity of additional hardware.

Our approach is based on reducing the area-time product by using hybrid algorithms **based on** a combination of Fast Hartley Transform (FHT) and DHT, i.e., by using higher radix FHT. To see this, let us consider again the DHT as

$$h = Hg$$

where

$$h = [h(0), h(1), h(2), \cdots, h(N-1)]^t$$

$$9 = [g(0), g(1), g(2), \cdots, 9(N-1)]^t \quad (26)$$

Let us also consider the case where M = N/2. The Decimation-in-Frequency (DIF) of a FHT of size N in terms of DHT of size N/2 is given as

$$\begin{matrix} h_e \\ h_o \end{matrix} = \begin{matrix} H(N/2) & H(N/2) \\ S(N/2) & -S(N/2) \end{matrix} \begin{matrix} g_p \\ g_r \end{matrix} \quad (27)$$

where

$$g_p = [g(0), g(1), g(2), \cdots, g(\frac{N}{2} - 1)]^t$$

$$g_r = [g(\frac{N}{2}), g(\frac{N}{2} + 1), \cdots, g(N)]^t$$

and

$$h_e = [h(0), h(2), h(4), \cdots, h(N-2)]'$$

$$h_0 = [h(1), h(3), h(5), \cdots, h(N-1)]^t$$

Also, we have

$$S(N/2) = H(N/2)K(N/2) \quad (28)$$

$$K(N/2) = \mathrm{Diag}\{\cos(2\pi K/N)\} + \mathrm{Diag}\{\sin(2\pi K/N)\}P \quad (29)$$

In the above expressions $H(N/2)$ represents a DHT of size N/2, and P is the permutation matrix as defined before. Then, Eq. (27) can be written as

$$\begin{matrix} h_e \\ ho \end{matrix} = \begin{matrix} H(N/2) & H(N/2) \\ H(N/2)K(N/2) & -H(N/2)K(N/2) \end{matrix} \begin{matrix} g_p \\ g_r \end{matrix} \quad (30)$$

or, equivalently

$$\begin{matrix} h_e \\ h_o \end{matrix} = \begin{matrix} H(N/2)(g_p + g_r) \\ H(N/2)K(N/2)(g_p - g_r) \end{matrix} \quad (31)$$

The flowgraph of the DIF FHT is shown in Figure 5. Our methodology leads to a both time- and area-efficient realization of a DFT of size N. This is illustrated in Figure 6.

In summary, note that, compared to the direct approach, the number of required CCD/CID chips has been reduced **from 8 to 4 at** the cost of only two additional simple bit-serial adders. Also, since the CCD/CID chip **has** a bit-serial data input format, performing bit-serial addition on the data will increase the computation time by only a cycle. Generalization to different M-to-N ratios is straightforward.

5. CONCLUSIONS

In this paper, we have presented the massively parallel CCD/CID hardware architecture. This new hardware technology leads to some interesting and fundamentally different results on the complexity of parallel computation. It also require a new framework **for the design and analysis of parallel** algorithms which **is drastically different from that usually** used **for** more conventional parallel hardware architecture. In particular, we would like to emphasize that on CCD/CID chips many operations can be performed in a time of O(1) which implies that the computation time is *independent* of the size of the problem. This is a drastic departure from classical theory of parallel computational complexity wherein the bounds of both time and processors are a function of problem size.

The CCD/CID computing technology could be used in a wide range of applications, As an example, we have presented some algorithms for signal processing applications. However, a much wider class of problems, such as partial differential equations (PDES), for which CCD/CID technology can be efficiently used, are currently under investigation. It is important to note that the current CCD/CID technology offers a rather limited accuracy. We have recently developed some techniques to significantly improve the computational accuracy while preserving fundamental advantages of CCD/CID technology.

ACKNOWLEDGMENTS

This research was carried out at the Center for Space ~licroclct ronicsrl'et hnology, Jet Propulsion Laboratory, California Institute ofq'ethnology under contract with the National Aeronautics and Space Adminstration.

REFERENCES

1. Neugebauer, **C. F., A.** Agranat, **and A.** Yariv, **U.S.** Patent 5,054,040.

2. Neugebauer, C. F., and A. Yariv, *Advances in Neural Information Processing Systems*, **4**, (in press, 1992).

3. Agranat, A., C. F. Neugebauer, and A. Yariv, *Applied Optics, 27, 4354* (1988).

4. Barhen J. et al, *Neuro-Nimes* '91, (EC2 publ., Paris), 73 (1991).

5. Barhen, J., N. Toomarian, A. Fijany, A. Yariv and A. Agranat *Neuro Nimes '92*, EC2 publ., Paris (in press 1992).

6. Agranat, A., Neugebauer, C.F., Nelson, R.D., and Yariv, A., *IEEE Trans. Circ. Syst.*, **37**, 1073 (1990),

7. Hockney, R. W. and C.R. Josshope, *Parallel Computers*, A. Hilgar publ., Bristol, UK (1981),

8. Schendel, U., *Introduction to Numerical Methods for Parallel Computers*, J. Wiley, Chichester, UK (1984).

9. Bracewell, R.N. *The Hartly Transform*, Oxford Univ. Press, New York, NY, 1986

10. Bracewell, R. N., Proc. IEEE, 72, 8, 1010, 1989

11. Bracewell, R.N., J. Opt. Sot. Am. 73, 12, 1832, 1983

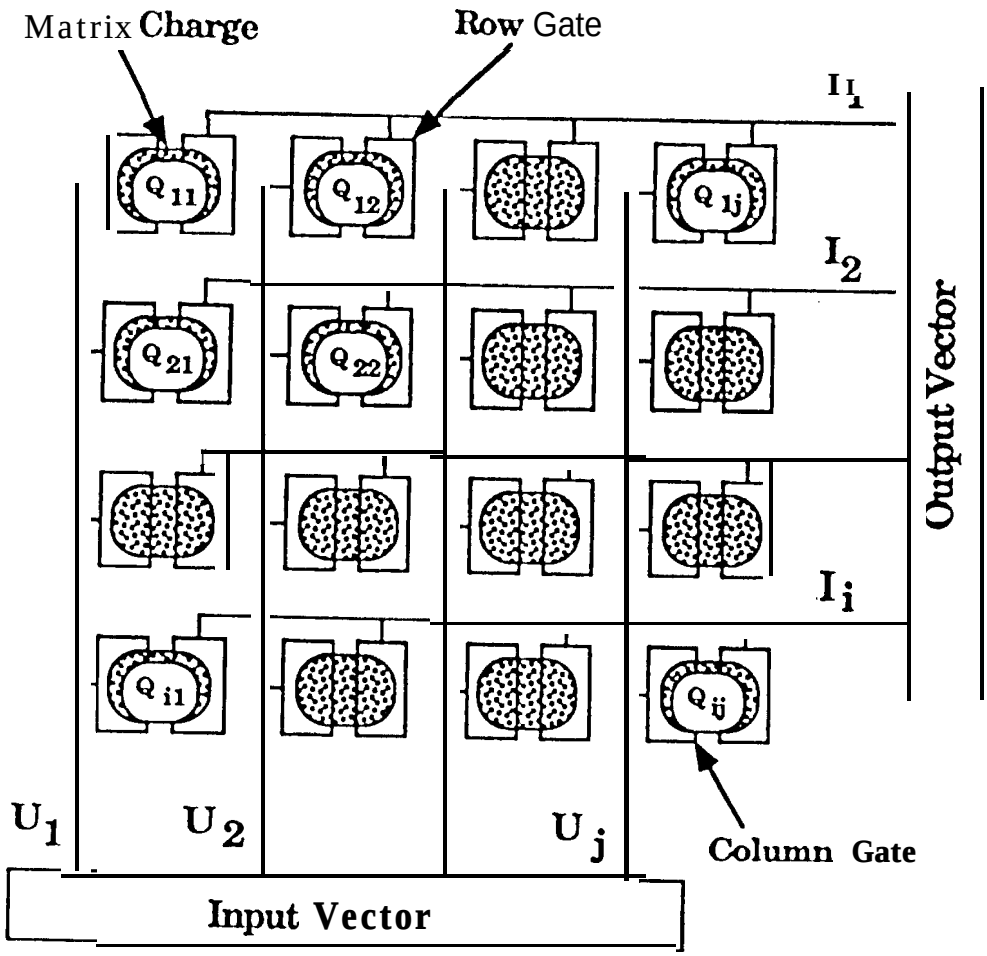


Fig. 1. The CCD/CID architecture consists of an array of CCD elements that are connected together by row and column **electrodes.** The matrix values are encoded as charge packets that sit underneath these gates in the silicon substrate, **The computation** occurs when the charges are transformed from the column gates to the row **gates which perform capacitive sum** operation.

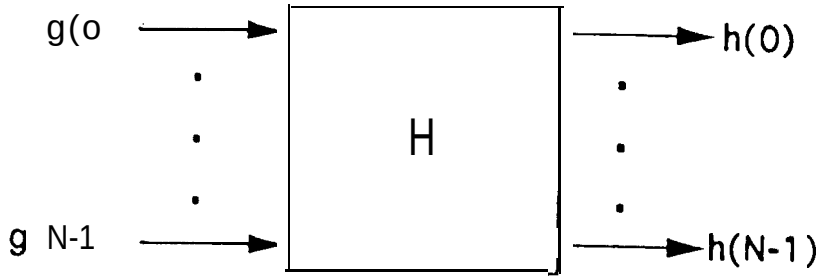


Fig. 2. The Schematic Kernel for the Discrete Hartley Transform

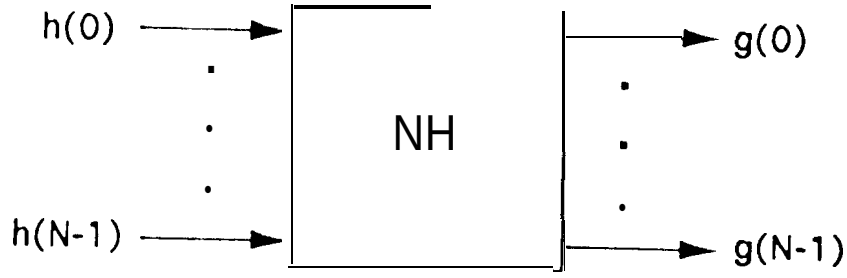


Fig. **3. The Schematic Kernel** for the Inverse **Discrete Hartley** Transform

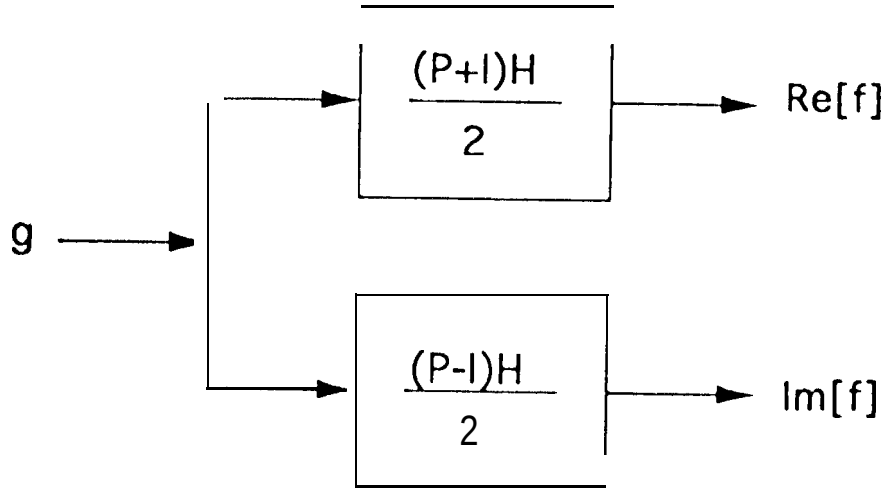


Fig. **4. Schematic Diagram** for Computation of DFT via DHT

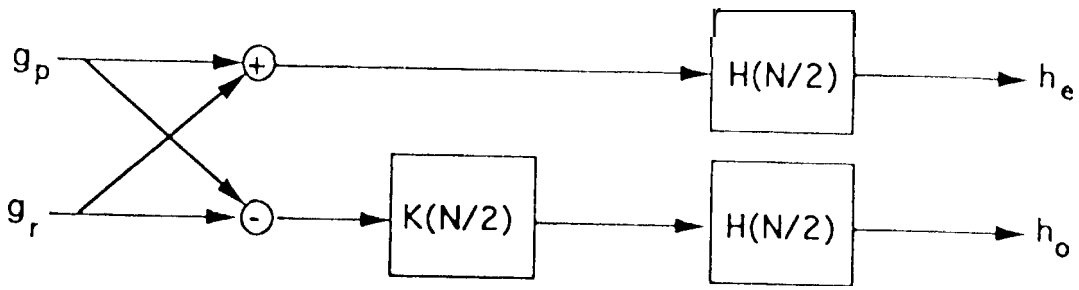


Fig. 5. The Flowgraph of Decimation in Frequency (DIF) of FHT

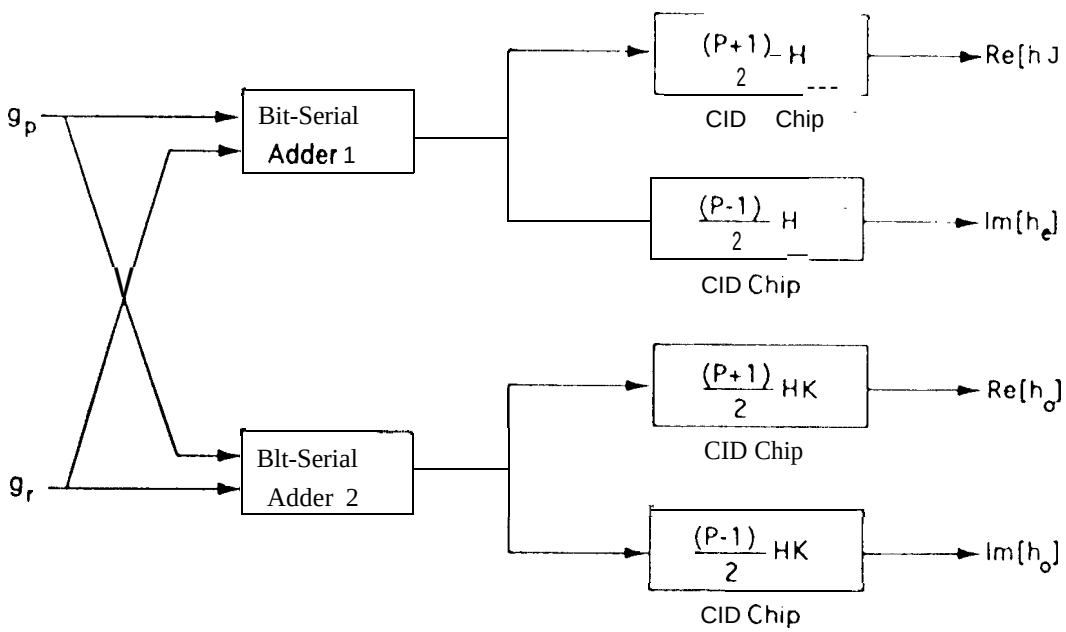


Fig. 6. Tine **and Area** Efficient Implementation **of Large** DHT